

How to read measurement data via Modbus on Socomec's Digiware metering system

When **energy** matters

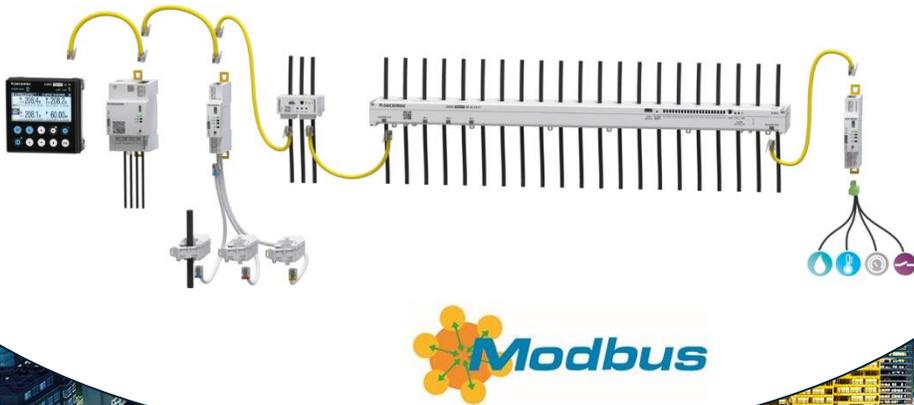


Table of Contents

Introduction	3
1. Prerequisites	3
1.1 Socomec Digiware concept	3
1.2 Socomec Modbus Register Maps	4
1.3 Zero vs. One Based Addressing	5
1.4 Supported Modbus Functions	5
1.5 Endianness / Word order.....	5
1.6 Data type description	6
1.7 Unit description.....	6
2. Principle diagram of the installation	7
3. Connecting to the Digiware system	8
4. Reading data on Modbus software for the Digiware I-31	8
4.1 Reading Energy kWh	8
4.2 Reading Power kW	10
5. Key takeaways	12

Introduction

In this technical note, we will explain how to read measurement data from Socomec power metering equipment using the Modbus Poll software, available for free as a 30-day trial version.

[Modbus Poll](#) is a full-featured Modbus® master program that can be used to monitor, configure, and test Socomec power meters. This document provides some pointers on using Modbus Poll successfully with the Socomec **Digiware** multi-circuit power monitoring system.

1. Prerequisites

1.1 Socomec Digiware concept

The Digiware metering system is a modular system where multiple meters are daisy chained via a proprietary bus called DigiBUS. Each device within the system holds its own unique Modbus slave ID.



Certain devices within the Digiware system can monitor multiple three-phase loads:

- The Digiware I-61 has (6) CT inputs and therefore can monitor (2) three-phase circuits.
- The Digiware BCM-2125 can monitor up to (8) three-phase circuits.

1.2 Socomec Modbus Register Maps

You can find the Modbus register map on the Socomec website under each product page.

For example, for the Digiware I-61 meter module, the Modbus communication tables can be found at the link below, then scroll down to the “**Documentation**” section and click on “**Communication table**”.

<https://www.socomec.us/en-us/p/diris-digiware-i>

The structure of the register map is as followed:

Functions: Read holding registers (03)

Address (decimal)	Address (hex)	Size (registers)	Description	Unit	Data type
18432	0x4800	1	Load status 0 : Disabled 1 : Enabled	-	U8
18433	0x4801	2	Date of last instance	s	DATETIME
18435	0x4803	1	Integration time	0.2 s	U16
18436	0x4804	2	Reserved	-	-
18438	0x4806	2	Reserved	-	-
18440	0x4808	2	Reserved	-	-
18442	0x480A	2	Frequency	mHz	U32
18444	0x480C	2	Ph-N Voltage : V1	0.01 V	U32

↓
Decimal Address

↓
Number of Words

↓
Measurement description

↓
Unit / scaling

↓
Data Type



The D-50/D-70 and M-50/M-70 do not have a Modbus register map, they are just acting as a system interface and communication gateway.



Digiware multi-load devices (I-61, BCM-2125 etc.) utilize a segmented Modbus register map. Each monitored circuit (Load) is assigned a dedicated register block; refer to the “Load 2” section of the register map to poll data for the second circuit.

1.3 Zero vs. One Based Addressing

Modbus is natively a zero-based addressing protocol (Reg+1), whereas JBus is a version of Modbus with a one-based addressing scheme (Reg+0).

In other words, to access Address n, in JBUS use address n and in MODBUS use address n+1. There is an addressing offset of 1 which must be managed.



*The register map from Socomec products is in Jbus.
Depending on the Modbus client used and its addressing scheme, +1 may need to be added to the register address to be able to read it correctly in Modbus.*

Example:

In the DIRIS Digiware system, the kWh (positive active energy) register in JBUS is 19841, to read it in MODBUS it will be 19842.



In Modbus Poll, register numbers are zero based by default (Modbus Poll already reads data in JBUS), so no +1 offset is needed.

1.4 Supported Modbus Functions

The supported functions by the SOCOMEC products are:

- Function 3: Read (Read Holding Register)
- Function 6: Write
- Function 16: Multiple write



In the Modbus.org standard documents, holding register addresses are given a prefix of “4” to distinguish them from other register types. For example, a holding register at address 1001 is referred to by “41001”. However, the leading “4” is not really part of the address.

Socomec registers do not include the Modbus function as a prefix, for example the register for total active power on a DIRIS Digiware I-31 is 18476.

1.5 Endianness / Word order

For the 16-bit register values, the most significant byte always precedes the least significant byte (in other words, the register value is transferred in Big-Endian byte order). This is because when the Modbus standard was created in the late 1970's, most processors used a Big-Endian memory architecture (where the most significant part of a multi-byte value is stored at a lower memory address).

However, the lack of standardization for values larger than 16 bits has resulted in a situation where Modbus implementers have to make an arbitrary choice as to which address of the register pair contains the most significant word of 32-bit values. Most programs for communicating with Modbus slaves can be configured for either register word order.

Socomec power meters follow the Big Endian format for 32-bit Modbus registers.

This means the high word (most significant) comes first, followed by the low word (least significant).

Example:

A Modbus read request for register 18476 (hex: 0x482C, representing *Total kW*) over two words returns 0001 73B5:

```
28/02/26 13:20:42.812 read @ 0x482c, size: 2
28/02/26 13:20:42.949 recv: 0001 73b5
```

- Register 13564 → High word: 0x0001
- Register 13565 → Low word: 0x73B5

Register 13564 contains the most significant word (0x001) and register 13565 contains the least significant word (0x73B5).

$0x000173B5 = 95,157 \text{ W}$.

1.6 Data type description

- U16 → Unsigned on 16 bits
- S16 → Signed on 16 bits
- U32 → Unsigned on 32 bits
- S32 → Signed on 32 bits

1.7 Unit description

Socomec power meters include registers with different units and scaling factors depending on the measurement type.

Example 1: Voltage

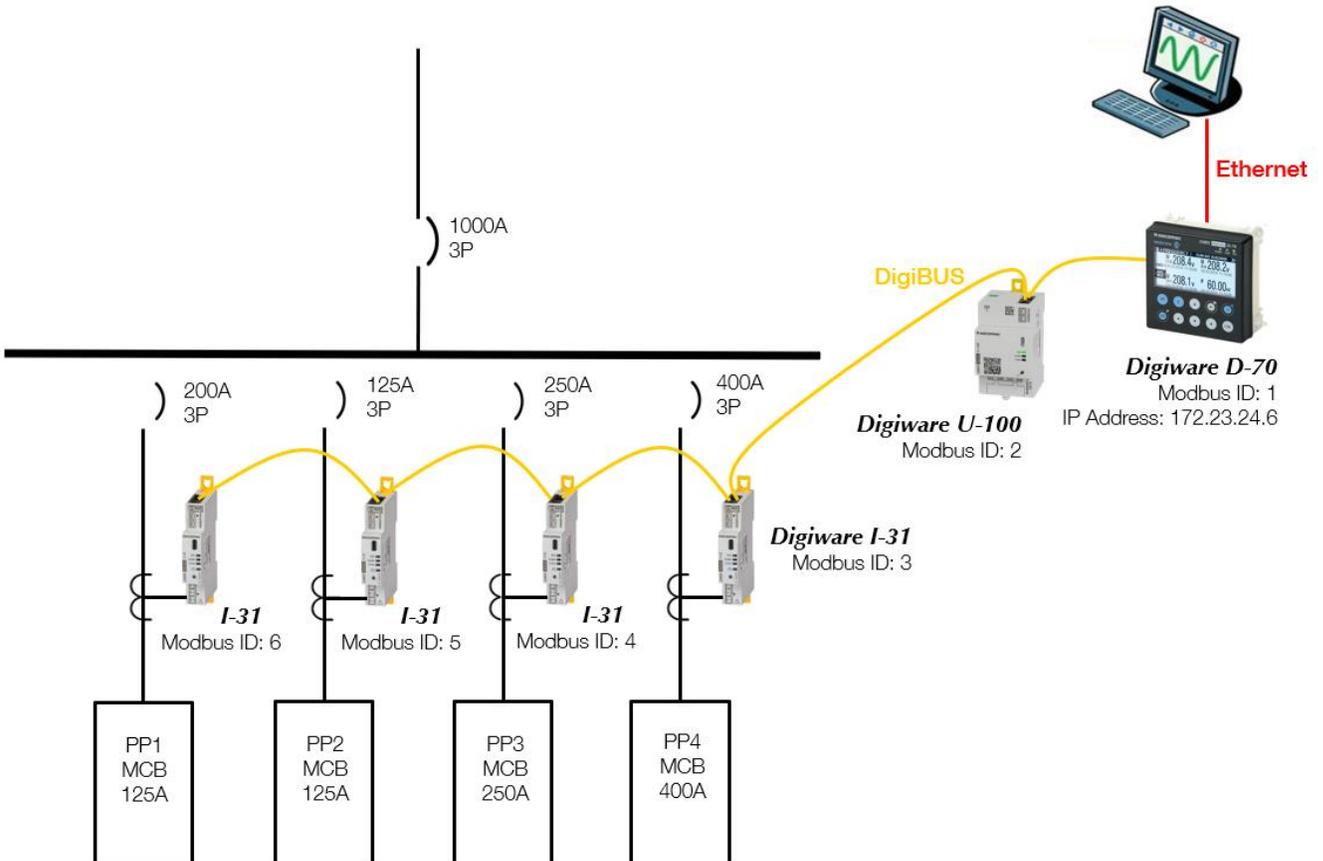
- Unit: V/100 (0.01 V)
- If Modbus returns **27,435**, the actual voltage is:
 $27,435 \div 100 = 274.35 \text{ Volts}$

Example 2: Current

- Unit: A/1000 (mA)
- If Modbus returns **400,435**, the actual current is:
 $400,435 \div 1,000 = 400.435$ Amps

➔ Now that we've covered the prerequisites for interpreting measurement data via Modbus on Socomec meters, let's look at a real-world example.

2. Principle diagram of the installation

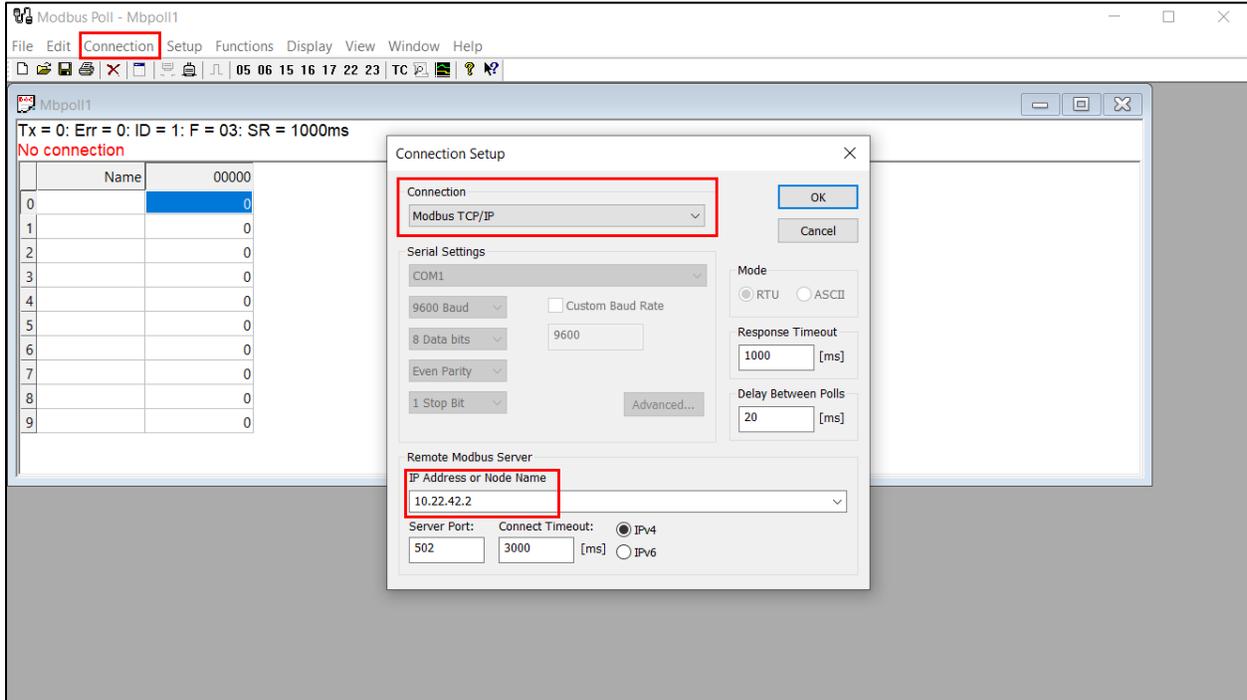


The Digiware M-50/D-50 or M-70/D-70 is just the gateway to system and doesn't have Modbus register maps with measurement. To access measurements, each Digiware I module has to be interrogated individually, each at its own Modbus slave ID.

3. Connecting to the Digiware system

In the Modbus Poll software, click on the “**Connection**” tab and click on “**Connect**”.

- Choose Modbus TCP/IP.
- Enter the IP Address of the Digiware M-50/M-70 or D-50/D-70.



4. Reading data on Modbus software for the Digiware I-31

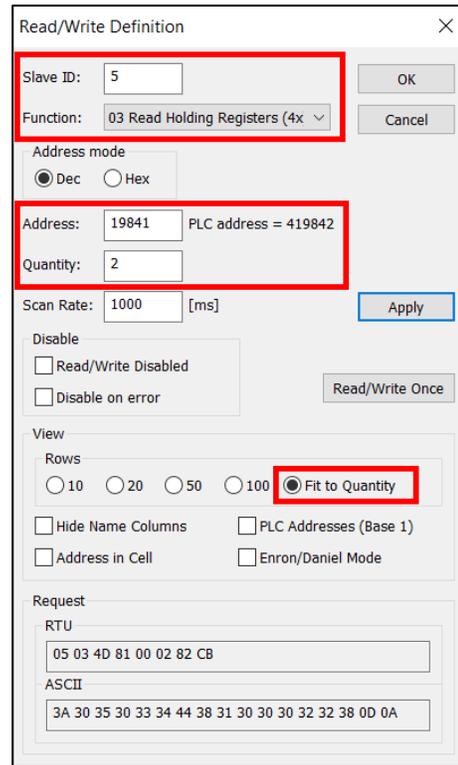
4.1 Reading Energy kWh

The register address for the kWh energy readings in the Digiware I-31 meter is given below. The energy value is an unsigned 32-bit integer, and the unit is in kWh.

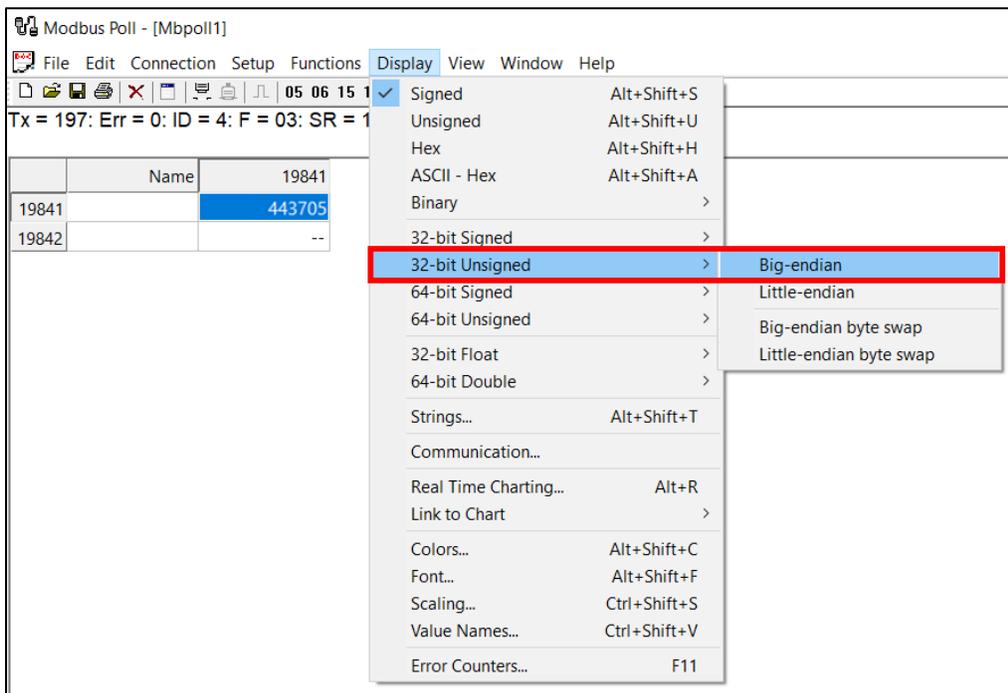
Address (decimal)	Address (hex)	Size (registers)	Description	Unit	Data type
19841	0x4D81	2	Total Positive active Energy: Ea+	kWh	U32

In Modbus Poll, click on the “**Setup**” tap, then “**Read/Write Definition**”.

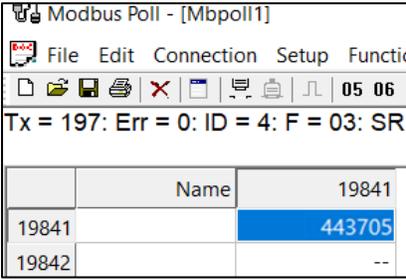
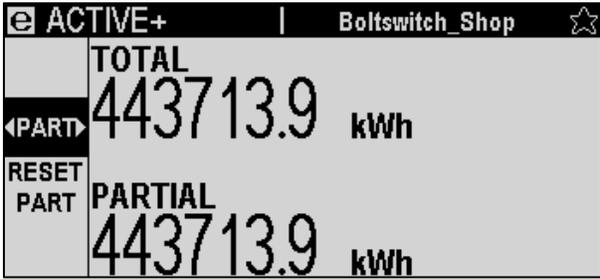
- Enter the Modbus Slave ID of the Digiware I-31 (5 in our example).
- Select Modbus Function 03 (Read Holding Registers).
- Enter the register address (19841).
- Under Quantity, enter the Number of Words (2).
- We also recommend the “**Fit to Quantity**” option if you don’t have too many rows in one register block.



Under “**Display**”, under “**32-bit Unsigned**”, click on “**Big-endian**”.



The decimal value returned in Modbus Poll is 443,713 kWh which matches the D-70 screen.

<p>Modbus Poll:</p>  <p>Modbus Poll - [Mbpoll1] File Edit Connection Setup Function Tx = 197: Err = 0: ID = 4: F = 03: SR</p> <table border="1"> <thead> <tr> <th></th> <th>Name</th> <th>19841</th> </tr> </thead> <tbody> <tr> <td>19841</td> <td></td> <td>443705</td> </tr> <tr> <td>19842</td> <td></td> <td>--</td> </tr> </tbody> </table>		Name	19841	19841		443705	19842		--	<p>A-200 screen:</p>  <p>ACTIVE+ Boltswitch_Shop TOTAL 443713.9 kWh PARTIAL 443713.9 kWh</p>
	Name	19841								
19841		443705								
19842		--								

4.2 Reading Power kW

The register address for the total three-phase kW power in the Digiware I-31 is given below. The power value is a signed 32-bit integer, and the unit is in W.

Address (decimal)	Address (hex)	Size (registers)	Description	Unit	Data type
18476	0x482C	2	Total active power	W	S32

In Modbus Poll, click on the “Setup” tap, then “Read/Write Definition”:

- Enter the Modbus Slave ID of the Digiware I-31 (5 in our example).
- Select Modbus Function 03 (Read Holding Registers).
- Enter the register address (18476).
- Under Quantity, enter the Number of Words (2).
- We also recommend the “Fit to Quantity” option if you don’t have too many rows in one register block.

Under “Display”, under “32-bit Unsigned”, click on “Big-endian”.

The decimal value returned in Modbus Poll is 20,756 W or 20.7 kW which matches the D-70 screen.

Modbus Poll:			D-70 screen:	

5. Key takeaways

- Socomec registers are in **Jbus**. A +1 offset may need to be applied in your Modbus client software if it's using a zero-based convention.
- Socomec registers are listed **without the Modbus function prefix** (example kWh register on Digiware I-31 is 19841).
- Socomec measurement registers are **Read Holding Integer** registers using **Modbus function 3**.
- Socomec 32-bit registers are in **Big Endian** (most significant word comes first).
- The D-50/D-70/M-50/M-70 are not meters, they are simple communication gateways and therefore do not have a Modbus register map. Refer to Modbus register maps of **Digiware I-xx** and **Digiware BCM** meters.
- The same Digiware meter can monitor multiple loads which is reflected in their Modbus register map organized in multiple "Load" sections.